An Interop Publication

ConneXions®

The Interoperability Report

September 1995 Volume 9, No. 9

ConneXions —
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, a division of SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.
Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: **connexions@interop.com**

Subscription hotline: 1-800-575-5717
or +1 610-892-1959

ISSN 0894–5926

From the Editor

First let me apologize for the late arrival of your August issue. Due to a mixup at the mailhouse, this edition did not go out until the end of August. Normally, issues are mailed sometime in the middle of each month. I also must apologize for giving you the wrong number for our new subscription agency. The number I gave you will indeed get you to Seybold Publications, but to their editorial offices rather than customer service. For questions regarding your subscription, please call 1-800-575-5717 or +1 610-892-1959. In case I didn't make this clear last month: *ConneXions* continues to be published as always by Interop Company, a division of SOFTBANK Expos, it is only the customer service aspects that are handled by Seybold Publications.

Since the dismantling of the NSFNET backbone in April of this year, the Internet now consists of a number of *Network Access Points* (NAPs) where service providers and other networks exchange traffic. Several aspects of this new system have been described in previous articles in *ConneXions,* and this month we bring you yet another installment. The article is by Bill Manning and is entitled "The Routing Arbiter in the Post-NSFNET Service World."

In July we published an article on AppleTalk in our "Back to Basics" series. This month we bring you further details about the routing aspects of AppleTalk. This in-depth article is adapted from *Routing in Communications Networks,* a book that comes highly recommended. (See review in our June 1995 issue). The article is by Fidelia Kuang and Alan Oppenheimer.

If you subscribe to any Internet mailing list or regularly follow discussion on a USENET newsgroup, you have no doubt been the victim of "spamming" at one time or another. The flooding of mailing lists with unwanted and irrelevant traffic is not a new phenomenon, but it is perhaps becoming more common as the number of Internet users continues to grow exponentially. In a short essay, "Michael Underwood" (not his real name) suggests some possible remedies to spamming. As always, we invite your comments and opinions. Please send them to **connexions@interop.com.** This is also the address you should use to suggest ideas for future articles. We would like *ConneXions* to address your needs as much as possible and hope you will help us in that regard by letting us know what's on your mind.

If you find yourself struggling with a particular aspect of networking we might have the answer for you in one of our 101 back issues. For a complete index, see **http://www.interop.com** or send us e-mail with your request. We can either send you the index file electronically or in hardcopy if you prefer.

The Routing Arbiter in the post-NSFNET Service World

by Bill Manning, ISI

Abstract

The United States *National Science Foundation* (NSF) has funded the *Routing Arbiter* (RA) [1] to provide stable, coherent routing in the Internet. With the Internet doubling every 13 months (according to some measurements) this is not as easy as it might be. The problem is compounded by the withdrawal of the NSFNET Service and the proliferation of Internet Service Providers and exchange points. A brief view from the RA perspective is given with some attention to tools and techniques that will facilitate the continued growth of the Internet in size, features, and function.

Introduction

On April 30th, 1995, the NSFNET Service was terminated, ending a nine year era of explosive growth for the Internet. By some measures, the Internet has doubled every 13 months and is showing no signs of slowing down. The latest figures from the Internet Society, Texas Internet Consulting and Network Wizards, along with figures from the InterNIC back up this premise. Perhaps the single most stabilizing influence has been the NSFNET, with its *Policy Routing Database* (PRDB) and "default-free" transit Service. This stability has not been without cost. The increasingly commercial Internet community has been concerned with the enforcement of the NSFNET *Acceptable Use Policy* (AUP) and the resultant breaks in reachability.

These facilities have been replaced with commercial services for transit, exchange points or *Network Access Points* (NAPs) for peering, and the Routing Arbiter. The Routing Arbiter has as its charter the continued maintenance of stable, unbiased global routing. To meet these tasks in the short term, the RA team, consisting of ISI and Merit, has focused on a replacement for the PRDB, which is the *Routing Arbiter Database* (RADB). As an adjunct to the database, we have written tools that allow any Network or *Internet Service Provider* (ISP) to define and register their own routing policies. With a large number of providers and several exchange points the Internet community finds itself in a policy rich environment, with levels of complexity that did not exist in the NSFNET Service era. Use of these tools allows ISP policy expression to be codified as router configurations. In addition, the RA has deployed *Route Servers* at the NSF NAPs and other locations. [25]

NSF Solicitation

The U.S. National Science Foundation, in winding down its support of what has been one of the better examples of technology transfer, recognized that it needed to focus on support of High Performance Computing and Communications. To this end, it released a solicitation [1] for a number of interlocking elements: a very high speed backbone to link its supercomputer centers, places where this backbone would be able to communicate with the rest of the Internet and its service providers, and an entity to facilitate stable, scalable, global routing so the Internet can continue to grow.

The vBNS

The *very high speed Backbone Network Service* (vBNS) is a private backbone, originally specified to run at 155Mbps (OC3c) and connecting up the NSF supercomputer centers (Cornell, NCSA, SDSC, PSC, and UCAR). The NSF is retaining its acceptable use policy on this infrastructure, in that it is to be utilized for research and education use only. The supplier of the vBNS service is required to connect to the Internet at all of the exchange points specified by the NSF.

The NAPs

The NAPs are level 2 interconnect or exchange points. NSF awarded three priority NAPs and one non-priority NAP. The NAPs are located in New Jersey (Sprint), Washington DC (MFS), Chicago (Bellcore and AADS), and the San Francisco area (Bellcore and Pac*Bell). The NAP architectures are currently either a bridged FDDI/Ethernet hybrid or an ATM(OC3/DS3)/FDDI hybrid. An additional exchange point is being constructed to support the other U.S. Federal internets access to the Internet at the NASA Ames facilities. The architecture of exchange points are being replicated around the global Internet, with exchange points in Europe and Japan. At each of these exchange points in the U.S., commercial and private use internets and ISPs touch down to exchange routing information and to transit traffic. A recent review of the exchange points has shown that the single 45Mbps NSFNET Service backbone has been replaced with as many as nine U.S. wide ISPs running 45Mbps backbones. Some studies [2] have indicated that with the increased load, the NAP fabrics as currently designed will not support the load offered by these ISPs.

The Routing Arbiter

The Routing Arbiter component has the charter to establish and maintain stable, unbiased, global routing and to advance the art and state of routing technology. Our initial efforts have gone into the NSFNET transition support and positioning to have the capabilities to support an increasingly rich environment for policy expression and interconnection. The architecture describing this phase of the RA activities is found in [3] and will be explored in the next section.

Routing Arbiter elements

The RA, in its efforts to meet the requirements for stable, unbiased, and global routing have laid out the following architectural elements, which we believe will meet ISP needs and will support the growth in Internet services. The RADB, which is part of the total Internet Routing Registry, the Configuration and Analysis suite of tools, and the Route Servers form the implementation today. Other, less tangible activities are education, engineering, and research, so we can stay ahead or right on the growth curve.

The IRR and RADB

Internet Operations have become dependent on two types of registries, a delegation registry such as the InterNIC, RIPE/NCC, or APNIC, and a routing registry such as the PRDB or RIPE-81. [27, 28]

Delegation registries are tasked with the transfer of authority and responsibility to manage Internet Assets for the public good. They assign blocks of address space, AS numbers and DNS names. In doing so they track the points of delegation. Over the years they have discovered that it is no longer feasible to maintain a monolithic registration service and expect it to scale. Three examples illustrate this;

- The migration from a flat host file to the Domain Name System
- The use of distributed NICs by region [4]
- The deployment of the Rwhois Service [5]

For the last few years, the NSF policy routing database was authoritative for general Internet traffic transit. However, with the increase in the number of public exchange points, it is no longer feasible to presume that this registry would scale. The RIPE staff recognized this problem as the infrastructure grew richer in Europe and they created the first public routing registry description and software [6]. Experience with this initial release led to refinement. Refinement brought it to the point that it was deemed appropriate to try and reconcile discrepancies between the PRDB and the RIPE registry for production use.

continued on next page

Routing Arbiter in the post-NSFNET World *(continued)*

The results of these efforts by RIPE and Merit have resulted in the RIPE-181 database and policy description. This release was stable enough that it was also released for general use in the Internet [7] and the code was widely distributed. ISPs with active registries based on the RIPE-181 code are the RIPE, the RA, MCI, CA*net, and others. At a meeting at the San Jose IETF (December 1994), representatives from these groups met and agreed that the collective information represented in these databases would be referred to as the *Internet Routing Registry* (IRR) and to ensure that the information was replicated, they would exchange the information on a periodic basis.

It was from this unified base that Merit selected its initial database. There were and are a series of problems related to the widespread use of the RIPE-181 registry. The problems we know of today are related to data duplication and directory synchronization. These are being addressed within the Routing Policy System working group [8] in the IETF. Until there is a resolution of these concerns, the RA team, in an effort to support unbiased access has adopted the view that the RADB is and can be considered a route repository of last resort. Anyone is free to register attributes within the RADB.

Once the base was selected, a thorough review was done of the database and the policy language to ensure that it could accurately and unambiguously represent the routing information and policies that were requested by ISPs. ISI was able to identify several inconsistencies with the RIPE-181 policy language and database [9] and has provided feedback to the community on changes that have been made in the RADB to support accurate representations of desired policies.

While this analysis was being undertaken, parallel efforts by Merit were proceeding to migrate the data from the old PRDB to the RADB. Perhaps the most difficult part of this effort was and is the ongoing need to retrain people to use the new registration procedures and tools. Although the tools have a common heritage [10] they must be tuned to a specific registry. ISPs must be aware of the subtle differences in tools between the PRIDE tools and the RPS updates to the PRIDE tools. It is important to note that for the RADB and the IRR in general, the intent is to place control of routing announcement in the direct hands of the Internet Service Providers and their clients. From a scaling perspective, a routing registry can no longer be run in a monolithic fashion with human intervention at every step. An added benefit is that with Internet users creating their own routing policies in the IRR there is less chance of bias or preferential treatment being injected by the RA or any operator of a component of the IRR.

Current directions on how to register in the RADB can be found in:

```
http://www.merit.edu/routing.arbiter/RA/RADB.tools.docs.html
```

Configuration and Policy Analysis

Since registration in a routing registry is usually an extra step, ISI has provided ISPs with tools to provide them with direct operational advantage for the effort of registration. This advantage is in the form of auto-configuration tools which build router configurations. These tools are able to evaluate policy expressions based on the RIPE-181 or the proposed RPS formats and generate router configuration files based on the outcome of the evaluations. The RA team utilizes these tools to generate configurations for the fielded route servers. CA*net modified the tools to generate Cisco router configurations that they use internally.

Merit used these tools to maintain the NSFNET Service router configurations in the last few weeks of its life. We would like to thank ANS for the use of their network to test out yet another configuration file format. The RA team realizes that as needs change, this toolkit will need to be upgraded.

Current directions on how to get the *RTconfig* Toolkit can be found in:

```
http://info.ra.net/div7/ra/
```

The current interface to the RADB is through e-mail. This constraint effectively limits the tools available today to essentially batch processing. The RA team recognizes that there are problems with this approach and realizes the need to have more interactive tools, such as a *telnet* interface to the RADB, as well as some "what-if?" tools to allow an ISP the ability to explore reachability options before committing changes to the RADB. Such tools are being developed now.

In addition to the development of these new tools, the RA team has picked up the PRIDE tools and is porting them to support RIPE-181 and RPS formats.

RSd

The target for all these efforts is to support configuration of routers. To show proof of concept and to add value to the NAPs, the RA has deployed route servers at each NAP. Given that the NAP traffic load is expected to be high, and that ISP routers would best be able to use their memory and cycles forwarding packets, the route server code was designed to compute a unique, composite view of the Internet on a per–peer basis. This is a novel change in router design and use.

The end result is that the NAP fabric can be viewed as a router system bus, with the ISP routers as the interfaces and the route server and the forwarding table computation engine. This design could be modified [11] to incorporate the separation of the control channel (routing updates) from the data channel (packet forwarding). To do this would allow better tuning of required bandwidths needed by the exchange point parties. To achieve this design, the RS software was adapted from the GateD Consortium's *GateD* version 3.5; we made extensive modifications to *GateD* to support per-ISP routing tables. [26] A number of releases have been made, with each successive release incorporating either functionality requested by service providers (e.g., correct handling of the *Border Gateway Protocol* (BGP) *Multi-Exit-Discriminator* (MED) attribute, knobs to configure the insertion of the Route Server's *Autonomous System* (AS) number in advertised AS paths) or those requested by RA team members (e.g., binary dumps of the routing tables).

Assuming the Internet will continue to double every 13 months encourages us to ensure that the choices we make will be viable at least for the short term. ISI has rigorously derived Route Server behavior from a formal characterization of the behavior of BGP speaking routers. This work [12] analyzes the storage requirements of Route Servers and suggests ways in which these storage requirements may be reduced.

This work has also led us to a complete redesign of our Route Server software. The new design reduces Route Server storage requirements (by more than an order of magnitude in some cases) by trading off some processing for lesser storage. Since the resulting implementation is significantly different from *GateD,* and is designed and optimized for Route Servers specifically, we have labeled this software RSd (for *Route Server daemon*).

continued on next page

Routing Arbiter in the post-NSFNET World *(continued)*

Work is also currently underway to design more efficient policy filtering in Route Servers. This is driven by the emergence of the need for more fine-grain policy; this need implies that policy filtering could become a dominant component of routing update processing in Route Servers. This improved design will be implemented in a future release of RSd.

Current directions on how to get the RSd software can be found in:

```
http://info.ra.net/div7/ra/
```

Operations and management

Placement of the route servers presents a number of interesting challenges. Since they are effectively stand-alone devices that may be unreachable, they have acquired many of the characteristics of intermittently visible devices.

Merit has deployed custom software into each route server that collects performance statistics, delay matrix measurements, and throughput measurements. This software discovers the state and topology of the NAPs once a minute, and automatically configures itself and its *ping* daemon to monitor all peers and peering sessions. Alerts such as "peer not reachable" or "peering session down" are generated and stored in a *Problem Table*. Both the *Network State Table* and the Problem Table are externalized via a bilingual SNMPv1/v2 agent.

As far as the RA team is aware, this is the first deployment of SNMPv2 technology in an operational environment [13], and as such, a number of problems have been found. These problems have been conveyed to the appropriate IETF working group for discussion. Primarily, configuration of the security features of SNMPv2 have proven to be difficult.

The Route Servers have also been configured to collect and store NAP performance statistics as seen from the Route Server. These statistics include:

- Interface statistics (in/out packets, in/out bytes, in/out errors)
- IP layer statistics
- BGP layer statistics

Other statistics collected include measurements of delay and packet loss characteristics between the RS and its peers, and throughput performance between each RS and an RA data collection machine.

The RA team has had a number of requests for additional reports that the ISP community is interested in for routing analysis. The RA has begun collecting and formatting data to present these types of information to the Internet community:

- Frequency of route flaps
- Total number of routes
- Aggregation statistics
- Stability of Route Server BGP sessions
- Volume of BGP updates

Other reports will note who is peering with the RS at various NAPs, how frequently the RADB is updated and configurations are run, and the stability of routing at the NAPs. The RA team will carefully consider privacy issues when making these reports available to the Internet community.

Education and engineering

The RA team is active in a number of forums where operational, research and administrative issues are discussed. There has been active participation in the routing designs at the hybrid NAPs and in outreach to the Internet Community. We encourage the formation and ISP participation in Operations forums like the *North American Network Operations Group* (NANOG) which is coordinated by Merit. RA information is available from two servers:

```
http://www.merit.edu/routing.arbiter
```

and

```
http://info.ra.net/div7/ra
```

Futures and research

The RA team is committed [14] to ensuring that the Internet continues to have stable, consistent, global routing with a goal for end to end reachability. In the current environment, we believe that the best way to reach this goal is through the use of the elements set forth above. Short term requirements that need work are:

- Improving the user interface for the RADB
- Correcting distributed database problems
- Releasing better "what-if?" analysis tools
- Speeding up configuration generation time
- Adding support for increased NAP speeds
- Improving Route Server asset utilization
- Better tools for fine-grained policy filters
- Protecting the Route Servers from attack

Longer term, ISI and Cisco are investigating the requirements for routing in and with IPv6. The RA team has identified the following issues for concentration of our research focus in this area:

• *Detailed analysis of IDRP dynamics with diverse topologies and routing policies.* As IDRP is deployed more widely any constraints on route selection and policy expression must be articulated. Moreover the routing registry provides a unique opportunity to detect configuration problems. Among other issues we will investigate a) the introduction of new route selection methods such as Cisco's communities and symmetric-bilateral routing agreements, and b) the effect of addressing assignment practices. Particular attention will be paid to IDRP running in a Route Server (RS) supported context.

• *In conjunction with our IDRP analysis and development of the RS system, we will be pursuing work in the area of routing protocol testing and emulation*. The target environment for IDRP/BGP is too large to begin to fabricate in a laboratory setting. Emulation is one technique for realizing reasonable scale experiments. Moreover, it allows more realistic investigation of protocol interaction than is usually achieved in simulations. Such emulation methods could become a critical part of protocol design, in addition to testing.

• *IDRP is a very flexible and extensible path vector routing protocol.* Two extensibility issues in particular will require attention in the coming year. The first is how to practically deploy IDRP for IPv6. The second issue is how to use IDRP across ATM clouds. Cisco has proposed introduction of a new NHRP routing algorithm. We will investigate the tradeoffs associated with incorporating the desired functionality into IDRP, versus interoperating IDRP with NHRP and based on our conclusion focus the necessary design and implementation activities.

continued on next page

Routing Arbiter in the post-NSFNET World *(continued)*

Amidst all the disagreements regarding routing architectures, most researchers agree that some form of explicit routing will be needed to accommodate heterogeneous routing demands, driven by both policy and quality of service. During the coming year we will be testing and refining our design of explicit route construction techniques. Three route construction mechanisms are under investigation. The first is RIFs, mechanism that uses IDRP queries with specified filters to obtain information from RIBs. The second is a path-explore mechanism to invoke constrained IDRP route computations. The final technique will be based on link-state style computations using the Routing Registry database. These explicit routes will be usable in conjunction with *Source Demand Routing Protocol* (SDRP), *Explicit Routing Protocol* (ERP), and PIM, however only the SDRP design is complete. Therefore, we will also complete our ongoing analysis of ERP and PIM-SM based on explicit routes.

References

[1] CISE/NCR, "NSF 93-52 — Network Access Point Manager, Routing Arbiter, Regional Network Providers, and Very High Speed Backbone Network Services Provider for the NSFNET and the NREN™ Program," Program Guideline nsf9352, May 1993.

[2] J. Scudder and S. Hares, "NSFNET Traffic Projections for NAP Transition," NANOG Presentation, October 1994, URL:
`http://www.merit.edu/routing.arbiter/NANOG/Scudder-Hares.html`

[3] D. Estrin, J. Postel, and Y. Rekhter, "Routing Arbiter Architecture," *ConneXions,* Volume 8, No. 8, August 1994.

[4] RIPE NCC, "Delegated Internet Registry in Europe," RIPE-NCC F-2 Version 0.3, March 1994.

[5] S. Williamson and M. Kosters, "Referral Whois Protocol," RFC 1714, November 1994.

[6] J-M. Jouanigot et.al., "Policy based routing within RIPE," RIPE-060, May 1992, URL:
`ftp://ftp.ripe.net/ripe/docs/ripe-060.txt`

[7] T. Bates, E. Gerich, L. Joncheray, J-M. Jouanigot, D. Karrenberg, M. Terpstra, & J. Yu, "Representation of IP Routing Policies in a Routing Registry (ripe-81++)," RFC 1786, March 1995.

[8] `rps-request@isi.edu`

[9] C. Alaettinoglu, and J. Yu, "Autonomous System Path Expression Extension to RIPE-181," Technical Report, USC/ISI, March 1995.

[10] T. Bates, M. Terpstra, D. Karrenberg, "The PRIDE project directory," March 1994, URL: `ftp://ftp.ripe.net/pride/`

[11] P. Löthberg, "D-GIX design," Private communication, 1993.

[12] Govindan, R., Alaettinoglu, C., Varadhan, K., and Estrin, D., "A Route Server Architecture for Inter-Domain Routing," USC Technical Report # 95-603, January 1995.

[13] Merit Network Inc., "Routing Arbiter for the NSFNET and the NREN, First Annual Report," April 1995.

[14] USC/ISI, and IBM, "Routing Arbiter for the NSFNET and the NREN — Annual Report 1994," April 1995.

[15] Yu, J., Chen, E. and Joncheray, L., "A Routing Design for the Initial ATM NAP Architecture," *ConneXions*, Volume 8, No. 11, November 1994.

[16] Rekhter, Y., and Li, T. (Editors), "A Border Gateway Protocol 4 (BGP-4)," RFC 1654, July 1994.

[17] Yu, J., Chen, E. and Joncheray, L., "Routing for the Initial ATM-NAPs In the Presence of the Special ANS/NSFNET Attachment," *ConneXions*, Volume 9, No. 1, January 1995.

[18] *ConneXions,* Volume 3, No. 8, August 1989, "Special Issue: Internet Routing."

[19] *ConneXions,* Volume 5, No. 1, January 1991, "Special Issue: Inter-domain Routing."

[20] Radia Perlman, *Interconnections: Bridges and Routers,* Addison-Wesley, Reading, Massachusetts, 1992.

[21] Adams, A., "The MERIT Policy-Routing Configuration System," *ConneXions,* Volume 7, No. 2, February 1993.

[22] Estrin, D., Rekhter, Y., Hotz, S., "A Unified Approach to Inter-Domain Routing," RFC 1322, May 1992.

[23] Estrin, D., Rekhter, Y., Hotz, S., "Scalable Inter-Domain Routing Architecture," SIGCOMM 1992, Volume 92, No. 4, October 1992.

[24] Estrin, D., Li, T., Rekhter, Y., "Source Demand Routing Protocol: A Component of The Unified Approach to Inter-Domain Routing," *ConneXions,* Volume 6, No. 11, November 1992.

[25] Yu, J., "The RA Route Server Service Overview," *ConneXions,* Volume 9, No. 8, August 1995.

[26] Hallgren, M and Honig, J., "GateD and the GateD Consortium," *ConneXions,* Volume 7, No. 9, September 1993.

[27] Karrenberg, Daniel, "The RIPE NCC and the Routing Registry for Europe," *ConneXions,* Volume 7, No. 11, November 1993.

[28] Conrad, David, "The Asia Pacific Network Information Center: Present and Future," *ConneXions,* Volume 8, No. 7, July 1994.

[29] Stockman, B., "Global Connectivity: The Global Internet Exchange (GIX)," *ConneXions,* Volume 7, No. 11, November 1993.

[30] *ConneXions,* Volume 7, No. 5, May 1993, "Special Issue: Focus on Europe."

[31] Guy Almes, Peter Ford, and Peter Löthberg, "Proposal for Global Internet Connectivity," June 1992.

[Ed.: An earlier version of this article was presented at INET '95 and appears in the proceedings from that conference.]

BILL MANNING is currently working at ISI on the Routing Arbiter project. He has been involved with NSFNET regionals, doing network engineering at both SESQUINET, THEnet, and MIDnet. His additional interests are with distributed systems and high speed networking. He is a member of IEEE and ACM and participates in the Internet Engineering Task Force and the Intercontinental Engineering and Planning Group. His current address is: `bmanning@isi.edu`

AppleTalk Routing

by Alan B. Oppenheimer and Fidelia Kuang

Introduction

The AppleTalk [1] network system was designed from day one with the user as its primary focus. Although AppleTalk was always intended to be a full-fledged, general-purpose networking system, its linkage with the Macintosh computer provided its developers with additional, unique requirements. Unlike other network systems being designed at the time, the principal design element in AppleTalk was (and continues to be) ease of use.

Just as the Macintosh was designed in a world where command lines and magic incantations were the norm, the networking world in which AppleTalk was designed was no less user-hostile. Networks of the day included such standard elements as terminators (little resistors that had to be put on the end of network segments), hard-coded unique addresses that users had to type in to identify network nodes, and static routing tables. As developers of a network to be used with "the computer for the rest of us," the AppleTalk development team was burdened with figuring out how to develop the network for the rest of us.

As the team set out to define the first *plug-and-play* network system, it tried to avoid compromising the functionality or the future of the network in any way. The concept of dynamic node addressing, for instance, was coupled with such standard local-area networking concepts as network numbers and sockets. [6] And when it was determined that a dynamic naming system was needed to simplify the location of network services, that system was designed to work, not just in a local environment, but across a full AppleTalk internet. (In this article, unless otherwise specified, *internet* always refers to an *AppleTalk internet.*)

The first AppleTalk system, *AppleTalk Phase 1,* was in many ways too successful. Primarily because of the phenomenon that the Macintosh created, the uses to which AppleTalk was put quickly exceeded those of its design center. AppleTalk internets were anticipated, but the size of those internets was not. The need to support various data links was anticipated, but the degree to which AppleTalk would be used with other protocols on those data links was not. And so, just like the Macintosh, AppleTalk was forced to evolve.

AppleTalk Phase 2 was introduced in 1989. Its primary focus was enabling the creation of large AppleTalk internets, and better interaction with other protocols. Because the Macintosh and its associated network products such as the LaserWriter printer had succeeded so well, the limitations under which Phase 2 was designed were even more severe than those imposed on Phase 1. Not only did the developers have to continue and extend the plug-and-play simplicity of AppleTalk Phase 1, but they had to do so in a backward-compatible manner, at least for the end nodes. With an installed base of more than a million nodes at the time, such nodes had to be upgradable only as necessary. It was required, however, that all the routers on an internet be upgraded before users could take advantage of many of the Phase 2 features, and even this limited requirement slowed the adoption of Phase 2 in many of the larger sites.

AppleTalk Phase 2 addressed most of the then-current issues around local-area networking. Much larger AppleTalk internets could be built, and they were. Customers for the first time also could successfully build large, wide-area AppleTalk internets.

AppleTalk customers' desire to build large, efficient, multiprotocol WANs, however, exposed yet another set of issues that needed to be dealt with. These included the need to provide a much more efficient routing protocol on WAN links and to tunnel AppleTalk through other protocol systems, specifically TCP/IP. [5] Many customers also expressed a desire for inter-organizational connectivity such as that provided by the (TCP/IP) Internet.

Having learned a lesson from the resistance to upgrading all routers in Phase 2, instead of creating a Phase 3, Apple introduced, early in 1993, a series of backward-compatible enhancements to AppleTalk routing referred to collectively as the *AppleTalk Update-based Routing Protocol,* or AURP. AURP provided a number of enhancements to AppleTalk routing, especially in WAN environments, and required no changes to currently installed Phase 2 routers.

AppleTalk internetworking basics

An AppleTalk internet includes the basic pieces found in most network systems: *nodes, networks, network numbers,* and *routers.* AppleTalk, as originally designed in Phase 1, defined an 8-bit node ID and a 16-bit network number. When it became clear that 254 nodes on one network would be insufficient, for compatibility reasons, Phase 2 had to utilize these same 24 bits in a manner that would circumvent the 254-node limitation. Thus, like IP and subnets, [5] AppleTalk Phase 2 introduced *network-number ranges.* An AppleTalk network (other than one using the LocalTalk data link, [1] which continues to use a single network number) is now identified by a range of network numbers. For each network number in the range, 253-node IDs are available. By configuring network-number ranges, an administrator can thus specify the maximum number of nodes on an AppleTalk network.

Fundamental to AppleTalk's ease of use, and a key aspect of any plug-and-play network, is a system for *dynamic node address assignment.* Nodes dynamically choose their network-number/node-ID combination (referred to as the node's address) with help from the routers on their network. Apple developed and patented a set of techniques whereby a node dynamically acquires a unique address on a particular LAN, in the absence of any routers (AppleTalk LANs are fully functional even when no routers are present—AppleTalk has always avoided requiring any centrally administered entity). When routers are present, this dynamic address-acquisition technique is extended to choose an internet-wide unique address.

Nodes first choose a LAN-wide unique tentative node address using a special reserved range of network numbers that are never propagated off the LAN. Nodes then use this tentative node address to talk to a router and determine the range of network numbers that is valid for their network. Finally the nodes dynamically choose a unique node address within this range.

Equally important to AppleTalk's ease of use is its *dynamic naming service.* To simplify the user's experience in browsing and selecting internet services, AppleTalk introduced the concept of *zones.* A zone is a logical group of nodes on an internet. Like area codes, zones subdivide the internet into more human-manageable regions. Zones are assigned human-readable zone names, and naming services are always performed within a particular zone. In Phase 1, each AppleTalk network was assigned, through configuration in the routers, a single zone name. In Phase 2, each AppleTalk network (other than LocalTalk networks, which continue to have only one zone name) is now assigned a zone list of up to 255 zone names.

continued on next page

AppleTalk Routing *(continued)*

Zone names are not necessarily related in any way to physical locality, and the same zone name can be used on any number of networks—nodes with the same zone name on all such networks are grouped into the same zone (like an area code that was used in different states).

As part of their dynamic address assignment, nodes select their zone name from the list of zones available on their network. To handle the case where a node cannot or does not choose a specific zone from the list, one of the zone names in the list is designated the *default zone* for that network. All nodes that do not otherwise choose a specific zone name are placed in the default zone for their network.

Each zone on a non-LocalTalk network has an associated *zone-multicast address*. Once a node has chosen its zone, it registers to receive packets sent to the specific zone-multicast address associated with that zone. Zone-multicast addresses are used to significantly reduce the overhead associated with dynamic naming. As will be described, routers ensure that name lookups are sent only to the appropriate zone-multicast address.

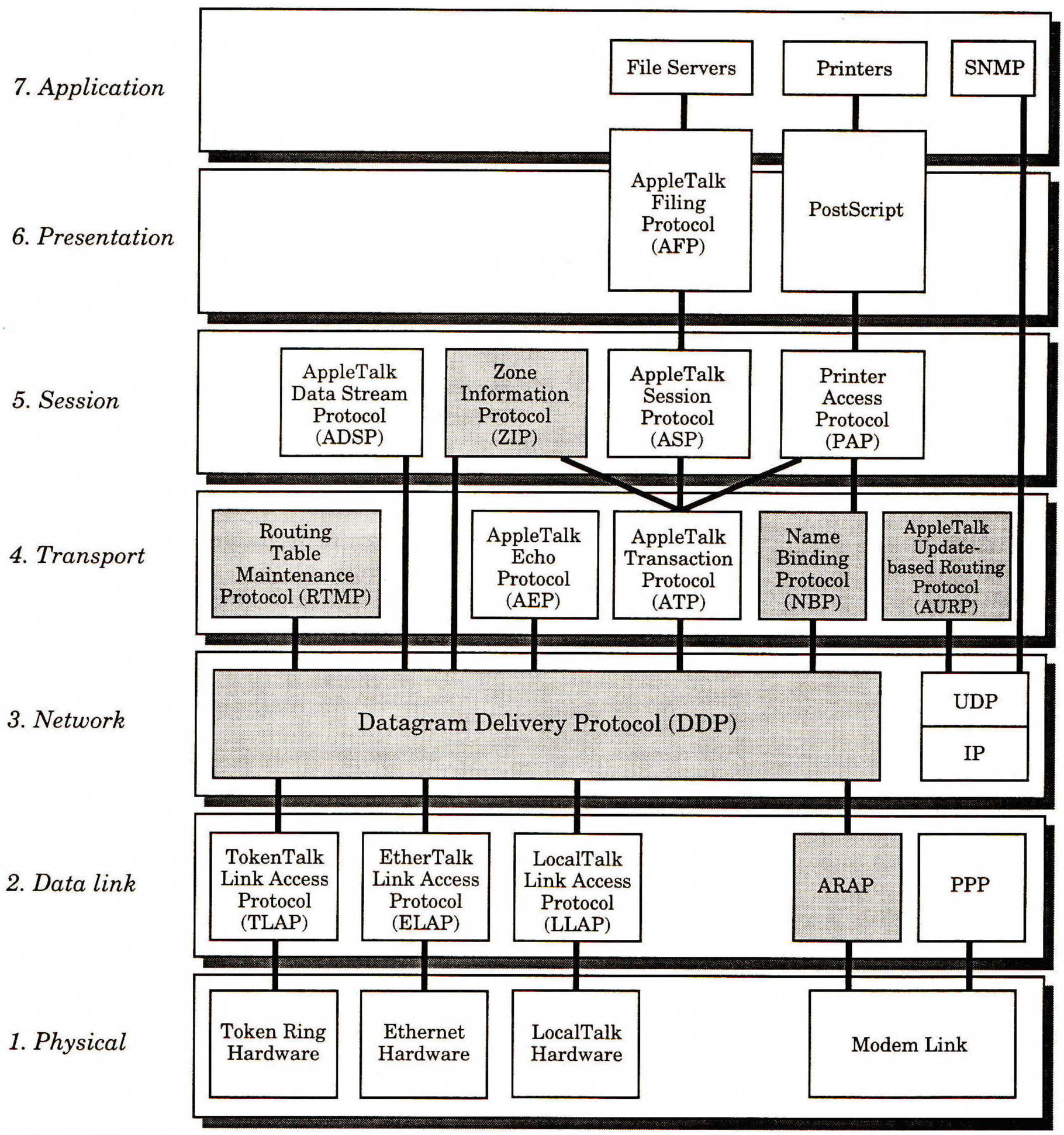

Figure 1: AppleTalk protocols mapped to the OSI Reference Model

Protocol summary

Figure 1 details the current AppleTalk protocol stack. The shaded protocols are discussed in this article. AppleTalk's data-link-independent network-layer protocol is the *Datagram Delivery Protocol* (DDP). DDP provides simple datagram service between two sockets on an internet.The *Routing Table Maintenance Protocol* (RTMP) is built on DDP and is responsible for propagating routing information on LANs throughout the internet. The *AppleTalk Update-based Routing Protocol* (AURP) is responsible for equivalent functionality on WAN links. The *Name Binding Protocol* (NBP) is responsible for the dynamic binding of user-visible names to network-layer addresses and is one of the principal contributors to AppleTalk's ease of use. NBP includes the concept of *zones* to greatly simplify service location for the user. Zone information is maintained in routers using the *Zone Information Protocol* (ZIP).

Because a network system is needed that is essentially as easy to use as the Macintosh itself, AppleTalk routing protocols have had to provide richer functionality than that in most present network systems. Plug-and-play networking does not come free. In addition to the standard features of propagating routing information and forwarding data packets, AppleTalk routing includes the ability to propagate the zone information needed for NBP, and to route the name-binding requests themselves. Finally, with the addition of AURP, AppleTalk provides the ability to build inter-organizational internets without the need for any sort of centralized administration. Each piece of AppleTalk routing is discussed in turn.

Propagating routing information

The Routing-Table Maintenance Protocol (RTMP) is used by AppleTalk routers to propagate network-reachability information. It is a classic distance-vector routing algorithm, much like the *Routing Information Protocol* [5] and includes *split horizon* to minimize the data sent in the routers' periodic multicasts. These multicasts are sent once every ten seconds. This rate supports the system's plug-and-play simplicity. AppleTalk networks (especially LocalTalk networks) tend to go up and down more often than those of most other network systems. AppleTalk routers multicast their routing information at a rate which generally results in alternate routes being adopted without tearing down of higher-level connections, and which also causes routes to new networks to be available quickly. Alternate route adoption is also aided by a technique referred to as *notify neighbor,* whereby routers indicate networks that have been aged out with a special tuple in their RTMP packets.

One of the most difficult aspects of setting up routers in an AppleTalk internet is configuring the routers with network-number ranges and zone information. To simplify configuration and minimize errors, the *seed router* was developed. A seed router is responsible for supplying (or *seeding*) the network-number range and zone list for a network. Only one router on any network need be a seed router. All others can be *nonseed,* meaning that they need not be configured with network information, and can thus be plug-and-play. More than one router on a network can be a seed router for redundancy, as long as all seed routers on the network are configured with the same seed information. Nonseed routers learn their network's number range by listening in on RTMP packets sent from seed routers. RTMP also provides the ability for both routers and end nodes to query for network information.

continued on next page

AppleTalk Routing *(continued)*

In addition to propagating routing information between routers, RTMP provides end nodes with just enough information to allow these nodes to communicate on an internet without configuration of any sort. Specifically, end nodes implement a small subsection of RTMP referred to as the *RTMP stub*. The RTMP stub is responsible for maintaining a node's network-number information, and for providing DDP with information needed to make forwarding decisions. Details of the RTMP stub's operation are provided in the section on AppleTalk data-packet forwarding.

Propagating zone information

The Zone Information Protocol (ZIP) is used by AppleTalk routers to propagate zone-to-network mapping information. ZIP is also used by end nodes, both as part of their dynamic node-address acquisition, and for enumerating the zones available on the internet.

The principal function of ZIP is to propagate the zone information associated with each network in an AppleTalk internet. ZIP works in concert with RTMP to perform this function. When an AppleTalk router obtains, through RTMP, the network-number range of a new network, the router uses ZIP to obtain the associated zone list of that network. The router first examines RTMP's routing database to determine the next router in the path to the new network. It then sends a ZIP query packet to that router, requesting the zone list for the network. If the receiving router has the full zone list, it sends the list in a series of ZIP reply packets. Otherwise it ignores the request (in this case the receiving router itself is still querying for the information).

A router's ZIP implementation periodically retransmits, to the next router in the path, requests for the zone information associated with any networks for which it does not have a complete zone list. (The number of zones in the zone list can be determined from any ZIP reply received for the network—each reply includes the total number of zones for the network.) Once a full zone list is received for a network, ZIP is free to propagate that information further by answering other routers' ZIP queries. Through ZIP, zone lists propagate outward from the associated network itself. Eventually, on a stable internet, every router will have the complete network-to-zone-list mapping of the internet. Such a list can be somewhat large, but the network operates much more efficiently with each router maintaining a full list (and the size of the list is almost always insignificant compared to the amount of memory used for forwarding buffers and the like).

ZIP monitors RTMP's routing database to determine when a network becomes unreachable. When a network is removed from the routing database, ZIP deletes the corresponding zone information. It does so to enable a network's zone list to be changed. Specifically, a network's zone list is changed by bringing the network down, waiting for its network-number range to be removed from the routing tables of all the routers on the internet (at which time the previously associated zone information is also deleted), and then bringing the network back up with a new zone list, which is then propagated throughout the internet by ZIP.

ZIP also plays a key role in a node's dynamic address acquisition. As part of this process, a node must select a node address including a network-number part that is within the network-number range of the node's network. The node must also select a zone name that is in the zone list for that network. Additionally, the node must be given the zone-multicast address associated with the chosen zone.

End nodes utilize the ZIP `GetNetInfo` command at startup time to determine information about their local environment. A ZIP `GetNetInfo` request is broadcast on the node's network and responded to by routers on that network. If the end node was previously active on the network, the `GetNetInfo` request contains information that the end node has saved from the last time it was active. Specifically, it contains the network number and zone name the node was using at that time. The routers, upon receiving the request, verify whether the information is still valid. If it is valid, they inform the end node with a `GetNetInfo` reply, and the node is able to continue the startup. If not, the reply contains the network-number range for the network, and also the name of the network's default zone (which the device can use until another zone is selected). The `GetNetInfo` reply also contains the node's zone-multicast address. The node registers on the multicast address so that it will receive zonewide NBP lookups sent to the zone-multicast address.

Routers independently calculate zone-multicast addresses using the algorithm specified by AppleTalk Phase 2. A zone-multicast address depends on the characters in the zone name and the specific data link on which the multicast packets will be sent (that is, zone-multicast addresses are different on Ethernet and token ring). The ZIP implementation in a router first converts the zone name to all uppercase characters. Then it hashes the string into a two-byte number using the same algorithm that DDP uses for performing checksum calculations. [1] Using the resulting two-byte hash value as an index, ZIP determines the corresponding zone-multicast address from the table of multicast addresses associated with the underlying data link.

Just as end nodes utilize a subset of RTMP to find out information about their internet environment, they also utilize a subset of ZIP for this purpose. Specifically, ZIP provides three functions that enable an end node to determine zone information for its local and global environment. These functions are implemented in simple request-and-response transactions, and utilize the *AppleTalk Transaction Protocol* (ATP) for this purpose. ATP is a transaction protocol that provides guaranteed delivery. ZIP requests are sent by an end node to any router on the node's local network. The three functions provided by ZIP for use by end nodes are:

- `GetZoneList` Returns a list of all zones in the internet
- `GetLocalZones` Returns a list of all zones on the sender's network
- `GetMyZone` Returns sender's zone name (used only on LocalTalk networks)

The responses for both the `GetZoneList` and the `GetLocalZones` requests list zones for the internet or for the sender's network respectively. Because the list of zones may not fit in one response packet, the end node may have to send multiple requests to the same router to obtain the whole list. Each request specifies a starting index for the zones to be returned in the response.

When an end node on a network chooses a new zone name (a rare event), it can obtain the list of zones for its network through the `GetLocalZones` request. After choosing a new zone name, the node must also configure itself to accept packets sent to the new zone's multicast address, which it can obtain with a `ZIPGetNetInfo` request.

continued on next page

AppleTalk Routing *(continued)*

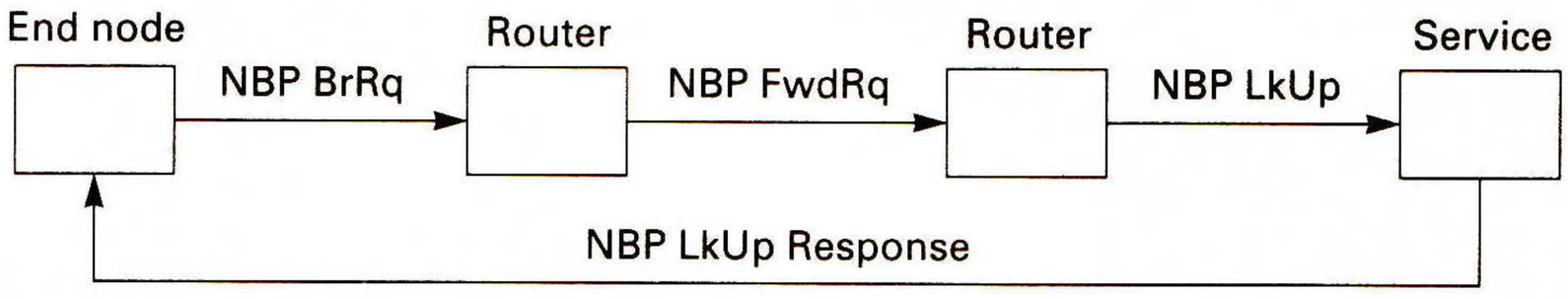

Figure 2: NBP packet forwarding

NBP packet forwarding

A router on an AppleTalk internet performs an important function in AppleTalk's dynamic name binding (Figure 2). The Name-Binding Protocol (NBP) enables the binding of network addresses to entity names. Name binding is an integral component of AppleTalk's service-location mechanism and of the ease with which it can be used. Because nodes obtain their network addresses dynamically, the network address of a service may change from time to time. The entity's user-visible name, however, changes infrequently. AppleTalk users generally locate services by name rather than network address.

An entity name consists of three identifiers: the object's name, the service type, and the name of the zone in which the service resides. An end node locates network services by performing a lookup on the service's object name and type in a specified zone. Wildcards are allowed in the name and type fields. The service or services that match the request respond to the requesting node. Through this exchange, the end node determines the network address of the service or services it specified by name. Routers on an internet facilitate the procedure by propagating the lookup request through the internet to all networks in the specified zone.

In an internet environment, an end node sends an *NBP Broadcast-Request* (BrRq) packet with the specified name, type, and zone to a router on its network. The NBP implementation in the router is responsible for propagating the node's request through the internet by converting the broadcast-request packet to *Forward-Request* (FwdRq) packets. The router sends one Forward-Request packet to a router on each network that has the specified zone in its zone list, utilizing the network-to-zone mapping maintained by ZIP.

When a router receives a Forward-Request packet, it is responsible for multicasting the request to all nodes in the specified zone on the destination network. It converts the Forward-Request packet to a *Lookup* (LkUp) packet. The router uses a zone-specific multicast rather than broadcast so that the packet interrupts only nodes on the network that are in the destination zone. To send a zone-specific multicast, the router addresses the lookup packet to the zone-multicast address corresponding to the destination-zone name. Nodes with a service matching the request respond to the original requesting node with the service's network address and completely specified service name. Notice that the BrRq, FwdRq, and LkUp packets carry the address of the original requesting node, which makes this direct responding possible.

AppleTalk data-packet forwarding

Both end nodes and routers have *data-packet forwarding* capability, end nodes to a limited extent and routers to a great extent. When an end node's DDP implementation sends a packet out on the network, the data-link-independent network-layer implementation in the end node looks at the destination-network number of the packet.

If the destination network is one that is directly connected to the node, the data-link layer sends the packet directly to the specified destination node. Otherwise, the data-link layer sends the packet to an internet router on the local network.

To enable this packet-forwarding capability, an end node keeps track of two pieces of information: the network-number range of its local network and the network address of an internet router on the local network. A node obtains this information through its RTMP stub, which listens to the RTMP packets sent by routers on its network. The node may send data packets to any of the routers on its network. Because the router picked may not be along the shortest path to the destination, however, it is possible for a data packet to travel an extra hop before reaching its destination.

To optimize the router selection, a node may implement the optional *best-router* algorithm, which enables the node to pick the best router for the destination of a data packet. A DDP implementation using the best-router algorithm looks at packets that come in from other, non-local networks and caches the source network number and the source data-link address, which indicate the address of the best router for that network number. The route through the best router should be the shortest route, by number of hops. When a node determines it needs to send a packet to a router for forwarding, the DDP implementation tries to find the destination network in its best router cache. If it succeeds, it sends the packet to the corresponding best router. Otherwise, it sends the packet to any known router.

Routers handle the bulk of data-packet forwarding in AppleTalk internets. A router manages any number of logical ports, which correspond to the different paths a packet may take. A router forwards packets through one of its ports from a source node or a previous router to the next router, or from a source node or a previous router to the destination node. The router keeps track of all reachable networks by listening to RTMP packets sent by other routers. The router stores information about each reachable network and the corresponding next-router address in internal routing tables. When the router receives a packet to be forwarded, it looks at the packet's DDP destination-network number. If the router can locate a routing-table entry corresponding to the destination network, it proceeds to forward the packet.

The routing-table entry contains information such as the network range, the network's distance in hops, the address of the next router, and the forwarding port. If the network's distance is zero, the destination network is directly connected to another port of the router. In that case, the router sends the packet to the destination node through the forwarding port specified in the routing-table entry. If the network's distance is greater than zero, the router ensures that the packet's hop count is less than the maximum, 15, before forwarding the packet. To forward the packet, the router increments the packet's hop count and sends the packet to the next router specified in the routing-table entry through the corresponding forwarding port.

AURP

The AppleTalk Update-based Routing Protocol (AURP) [2] is more than just a routing protocol. It encompasses a number of techniques for enhancing AppleTalk routing, especially in WAN environments. These techniques are completely compatible with Phase 2. The main techniques are the update-based propagation of distance-vector routing information and the tunneling of AppleTalk in other protocol systems such as TCP/IP.

continued on next page

AppleTalk Routing *(continued)*

These core techniques feature little or no routing traffic on a stable internet, as well as scaling from point-to-point modem links to large, wide-area internets. They also provide a specification of AppleTalk tunneling over IP with the same functionality as AppleTalk's LAN routing protocols, RTMP and ZIP. Other techniques defined in AURP include network-number remapping, clustering, hop-count reduction, and two security techniques: device hiding and network hiding.

Tunneling AppleTalk in other protocol systems enables an AppleTalk internet to span much wider areas. A *tunnel* is thus a virtual data link between two or more segments of an AppleTalk internet. (AURP supports the concept of multipoint tunnels.) An AppleTalk router that connects a local AppleTalk internet to a tunnel is referred to as an *exterior router.*

An exterior router continues to speak RTMP and ZIP on AppleTalk networks while speaking AURP across a tunnel (Figure 3). AURP remains compatible with Phase 2 by providing all the functionality of RTMP and ZIP on the tunnel. Exterior routers use AURP to exchange across the tunnel the same network information and zone information that RTMP and ZIP provide on AppleTalk networks. Exterior routers can use this information exchange to maintain the same routing tables and zone tables used by RTMP and ZIP. As a result, nodes on AppleTalk internets connected by a tunnel perceive no difference between AURP exterior routers and ordinary Phase 2 routers.

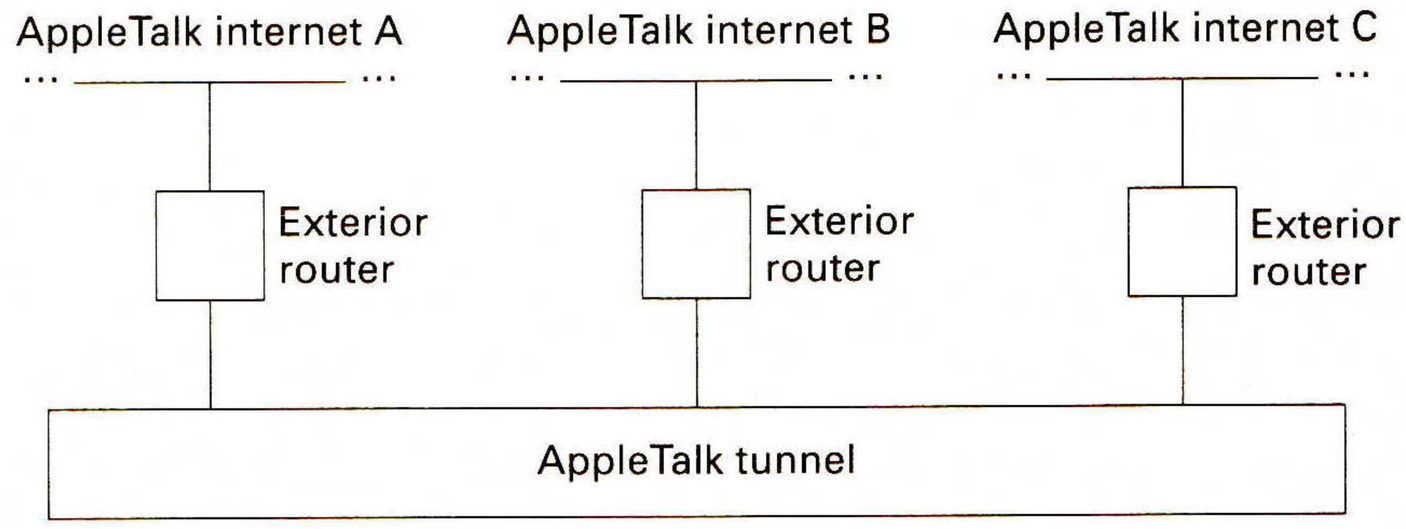

Figure 3: AppleTalk tunnel

Update-based routing

Because AURP is *update* based, exterior routers generally send routing packets only if the network topology changes. In contrast, RTMP routers send periodic multicasts regardless of any network changes. For exterior routers to maintain a view of the internet that is consistent with RTMP routers, the exterior routers depend on reliable connections among themselves. To simplify router discovery, an exterior router is generally configured with an address of at least one other known exterior router. Router discovery is thus static rather than dynamic.

After establishing a connection with another exterior router, the exterior routers initially exchange complete routing information for each router's local internet. As in RTMP and ZIP, the routing information includes both network ranges and zone lists. Once the initial exchange is complete, the routers have successfully established a tunnel connecting their AppleTalk internets. Because the routers maintain a reliable connection, they depend on each other to send reliable updates containing topology changes.

The minimal interval between sending of updates is a parameter that can be adjusted for a network's characteristics. If a router has no update information to send at an update interval, it does not send routing packets. As a result, stable internets have essentially no AURP routing traffic, compared to the periodic traffic of RTMP.

During the update interval, the exterior router buffers pending update events. Possible events include:

- Addition or removal of a network from the routing table
- A change in the routing path so that the router accesses the network through its local internet rather than the tunnel, or vice versa, through the tunnel rather than locally
- A change in distance to a network

At the end of the interval, an exterior router sends a routing-update packet if at least one change in routing information occurs during the update interval. An update packet may carry different types of update information for different network ranges, but it may include only one update event for a given network range. If multiple events occur for a given network during the interval, the router combines those events into one resulting event. For example, if a network is deleted from the routing table and then added back in, the resulting event is a distance-change event because the distance to the network may have changed. Buffering of events allows the router to smooth out fluctuations in the internet.

Tunneling in AURP

Because TCP/IP networks are prevalent, a goal for AURP was to specify a standard way of tunneling AppleTalk through TCP/IP, to take advantage of the infrastructure of TCP/IP installations. As a result, in addition to specifications for point-to-point tunneling, AURP specifies a method for multipoint IP tunneling using *User Datagram Protocol* (UDP) [5] packets. A datagram-oriented protocol, UDP provides a simple best-effort transport mechanism for AURP.

Exterior routers that connect AppleTalk internets through an IP tunnel act as routers on an AppleTalk internet and as hosts on the TCP/IP internet. The exterior routers use IP only for tunneling, that is, for propagating AppleTalk routing information and for forwarding AppleTalk data packets. AURP defines a simple transaction protocol on top of UDP to ensure reliable delivery of routing information. The routing protocol defines transactions for establishing a connection between exterior routers, for exchanging routing information, and for maintaining both the connection and the routing information. For data forwarding, an exterior router encapsulates AppleTalk data in UDP datagrams and sends the resulting packets across the TCP/IP internet to the next exterior router. The destination exterior router decapsulates the AppleTalk data and then forwards the AppleTalk packets to the destination AppleTalk network.

The AURP packet format includes the network system's headers (IP and UDP) followed by an AURP domain header followed by either AppleTalk data or AURP routing data. The current version of AURP specifies that a domain represents the exterior router's local AppleTalk internet. The domain header includes the destination- and source-domain identifiers and the packet type (data or routing). A domain identifier uniquely specifies an AppleTalk domain. Exterior routers currently use domain identifiers based on IP addresses.

continued on next page

AppleTalk Routing *(continued)*

Because the exterior router can derive its domain identifier from its IP address, it is not necessary to configure the domain identifier. Notice that UDP port 387 is reserved for use by AURP.

Network-number remapping

By allowing disparate AppleTalk internets to connect through a tunnel, AURP enables connectivity between organizations' internets that are likely to be administered independently. As a result, conflicting AppleTalk network numbers may exist. AURP specifies a technique for resolving network-number conflicts—*network-number remapping.*

A network administrator for an organization is aware of the network ranges used locally, but not necessarily those used by other administrators on the same tunnel. When configuring a port on an exterior router that supports remapping, the administrator can specify a range of network numbers that is not used locally. By reserving this range for network-number remapping, the exterior router can map the network numbers of networks imported over the tunnel into network numbers in the remapping range. The router represents these imported networks to the local internet as networks with ranges in the remapping range.

An exterior router internally keeps track of network-number mappings by keeping a table of the domain identifiers and the remote-network numbers from each domain. Together the domain identifier and the remote-network number make up a unique identifier for the imported network. Each unique identifier corresponds to a network number in the remap range, which the router uses to represent the network locally. The exterior router uses the unique identifier in incoming packets to identify the corresponding local-network number and conversely, the local-network number in outgoing packets to identify the corresponding unique identifier.

To achieve network-number remapping transparently, the exterior router is responsible not only for representing the imported networks in the remap range locally, but also for ensuring that AppleTalk data packets between the local networks and the remote remapped networks contain correctly mapped addresses. The remapping exterior router maps the network numbers in incoming packets into the remap range. Similarly, it maps network numbers in outgoing packets back to the actual network numbers. Specifically, the exterior router remaps network numbers in the DDP source-address field of incoming packets, the DDP destination-address field of outgoing packets, the NBP entity-address field in AppleTalk NBP packets, and the routing-data field in AURP routing-information packets.

Remapping of network numbers is possible because of AppleTalk's dynamic addressing capability. Remapping would not be practical to implement if exterior routers had to remap network numbers carried in the data portion of AppleTalk packets. However, because network addresses are dynamic, the data portion of a packet generally carries NBP entity names instead of network numbers. Dynamic addressing, developed to make AppleTalk easier to use, thus makes possible network-number remapping, which, in turn, further enhances AppleTalk's ease of use.

Because network-number remapping allows networks to be represented by different network numbers, some problems arise when routing loops are present across a tunnel. A routing loop is present when more than one path exists between two exterior routers.

One path is across the tunnel and another path may be through the local internets of the routers. After a remapping router maps an imported network into its local-remap range, the routing information for the remapped network propagates throughout the local internet. If a redundant path (a path other than the tunnel) exists, it is possible for that routing information to reach the exterior router that originally exported the network. Because the network number is remapped, the routing information appears to be new information to the exporting exterior router, which then exports the routing information across the tunnel. The original remapping router cannot determine that this new network number actually represents the same network that was previously remapped. The remapped network number appears to represent a new network, which is referred to as a *shadow network*. Because of the loop, each shadow network can be repeatedly remapped until the distance to the last shadow network reaches the hop-count limit.

To avoid infinite remapping loops, the exterior router should implement at a minimum the loop-detection techniques specified by AURP. Loop detection is essentially a two-step process: looking for loop indicative information and then verifying the presence of a loop. If a loop is verified, the exterior router is responsible for eliminating the loop by disconnecting from the tunnel and thus eliminating the path through the tunnel.

Loop-indicative information seems to refer to a network across a tunnel, but could in fact refer to a network in the local internet. Specifically, loop-indicative information is a network range of exactly the same size as a network directly connected to a port of the exterior router, and a corresponding zone list that is exactly the same as the zone list for the local network. If these two parameters--the network-range size and the zone list—match those of the directly connected network, it is possible that the local-network range was remapped and exported back to the router through a loop.

After detecting loop-indicative information, an exterior router verifies the existence of a loop by sending out a *Loop-Probe* packet, which carries unique information about the sending router so that the router recognizes the packet if it receives it. The exterior router sets the destination address of the packet to be the suspected shadow address of the port for the directly connected network. For example, if the port for the directly connected network is at network 100, node ID 128, and the remapping router remaps the network number to 1000, the destination-network address of the packet will be network 1000, node 128. The exterior router sends the packet through the tunnel to the router that provided the loop-indicative information. If a loop is present, the exterior router that sent the Loop-Probe packet receives the packet through its local internet. After verifying the unique packet data, the exterior router determines the existence of a loop and eliminates the loop. If a loop is not present, the exterior router does not receive its Loop-Probe packet. The exterior router should send a Loop-Probe packet at least four times with retransmission timeouts no less than two seconds. If it does not receive its Loop-Probe packet after four attempts, it determines that a loop does not exist.

Clustering

AURP specifies a *clustering* technique that allows exterior routers to represent multiple networks accessible through the tunnel as a single network range within its local internet.

continued on next page

AppleTalk Routing *(continued)*

Because a router generally tracks each existing network range in its internal-routing tables and sends information about each range in its RTMP routing packets, clustering reduces the size of routing tables and reduces routing traffic within a local AppleTalk internet. Remapping networks into a sequential range of network numbers enables an exterior router to represent a series of networks as one network. For example, if an exterior router is remapping networks into a range 1000–1500, such that there is a network 1000–1005, 1006, and 1007–1010, it can represent those networks as a clustered network with a range of 1000–1010. The zone list for the resulting cluster includes all the zones associated with any of the individual networks. Because a zone list for a network may never exceed 255 zones, a cluster may never have more than 255 zones in its zone list.

Just as in remapping, the exterior router is responsible for ensuring that packets between the local networks and the remote clustered networks contain correctly mapped addresses. The exterior router has the additional responsibility of monitoring NBP Forward-Request packets to be sent across the tunnel. From the DDP destination-network number in a Forward-Request packet, the exterior router determines the corresponding cluster. Rather than sending the packet to every network in the cluster, the router sends the packet only to networks whose corresponding zone list includes the zone name present in the Forward-Request packet. Because the router must determine the destination networks corresponding to the specified zone, it must maintain a list of the actual networks and a mapping of those networks to their corresponding zone lists.

Hop-count reduction

Normally, every router increments a packet's hop count when forwarding a packet. To prevent a packet from looping continuously, a forwarding router discards the packet when the hop count reaches the limit of 15. The AppleTalk hop-count limit constrains the diameter of an AppleTalk internet to 15 hops. (In AppleTalk's original design center, a maximum of 15 hops was sufficiently high for an AppleTalk internet.) When a tunnel connects AppleTalk internets, the diameter of the resulting internet may exceed 15 hops. To maintain full connectivity between the two segments of the internet, an exterior router may implement the AURP technique for hop-count reduction.

To allow a DDP packet to go beyond the hop-count limit, an exterior router may reduce the hop-count value in a packet received through the tunnel before forwarding the packet to the local internet. The exterior router reduces the hop count only by the amount necessary to allow the packet to reach its destination. Because the exterior router knows the distance to the destination network, it can calculate the maximum adjusted hop-count value. If, for example, the packet will reach its destination without hop-count reduction, the router just forwards the packet. If the packet's hop count plus the distance to the destination exceeds the limit of 15, the router reduces the hop count to a value calculated by subtracting the distance in hops to the destination network from 15. The adjusted hop-count value allows the packet to traverse the local internet and reach its destination.

In addition to reducing the hop count of DDP packets, a hop-count-reducing router is responsible for representing, within its local internet, reachable distances to the networks accessible through the tunnel. AURP specifies that the exterior router represent all networks through the tunnel as one hop away. Other routers on the local internet then perceive those networks to be reachable and maintain full connectivity.

With hop-count reduction, loops in an internet topology allow an errant packet to potentially circulate forever, for the hop count may never reach the limit. As a result, usage of hop-count reduction is recommended only in topologies without loops. Because AURP's loop-detection technique ensures there are no loops in a remapping environment, an exterior router performs hop-count reduction only when remapping is active and thus there are no loops.

Security

AURP specifies two types of basic security: *device hiding* and *network hiding*. Both types allow a network administrator to determine which services or networks should be visible to which portions of the internet. These hiding functionalities are particularly useful in wide-area environments in which different organizations connect their internets.

A network administrator can configure any AppleTalk Phase 2 router that supports device hiding to hide a particular device. The router keeps the specified device in its local internet from being visible to end nodes on the internet by filtering out NBP Lookup-Reply packets from that device. If nodes cannot receive the device's NBP replies, they cannot locate the service. That is, they cannot learn the device's name and they cannot associate its name with a network address. As a result, device hiding makes it difficult for nodes to access the hidden device. It does not, however, provide true device security because it does not limit direct network access to the device. A node can still send packets to a hidden device's dynamic network address, if known. Thus, device hiding provides a simple security technique that prevents locating services through NBP.

With network hiding, a network administrator can configure a tunneling port to hide routing information for networks being imported across a tunnel or to hide routing information being exported across a tunnel. By limiting imported routing information to information about specific networks, an exterior router reduces routing information that is maintained and propagated by routers on its local internet. Because the networks are not available to nodes on the local internet, neither are the corresponding zones and devices on the networks visible. Similarly, by exporting no information about hidden networks, an exterior router prevents other exterior routers from learning about the existence of those networks. Whether limiting imported or exported routing information, network hiding provides a technique for security across tunnels at the network level. By reducing the number of accessible networks, network hiding also reduces network traffic across the tunnel as well as in the local internets.

Network hiding prevents nodes on networks across the tunnel from accessing any devices on the hidden networks. Conversely, it prevents nodes on the hidden networks from accessing any devices across the tunnel (nodes cannot respond to packets from a hidden and therefore unknown network).

By employing network hiding, a network administrator can set up an internet area known as a *free-trade zone,* an area of the internet accessible by two other parts of the internet that cannot access each other. For example, two organizations may want to share information, but they may not wish to have direct connectivity. A free-trade zone allows both organizations to access a common area and to keep their internets isolated otherwise. A router that is creating a free-trade zone through network hiding exports only networks that are in the free-trade zone. Each organization can see the free-trade networks, but not the networks from the other organization.

continued on next page

AppleTalk Routing *(continued)*

Alternative AppleTalk routing methods

Besides AppleTalk routing through RTMP, ZIP, and AURP, other mechanisms can extend the user's reach with AppleTalk connectivity. Some of these methods require encapsulation of other protocols, such as TCP/IP, within AppleTalk or, conversely, AppleTalk within other protocols. Two such methods of providing gateways between DDP and IP, MacIP and IPTalk, [4] are described below. Another method, used in Apple Remote Access, allows remote nodes to access an AppleTalk internet over a point-to-point link.

Gateways between DDP and IP

Gateways are devices that translate between sets of protocols, such as AppleTalk and TCP/IP. Because TCP/IP connectivity is a basic requirement in the university environment, early university Macintosh users spearheaded work to allow integration of the Macintosh in their TCP/IP internets. Macintosh computers communicated using AppleTalk, and so the goal was to create a gateway (and client software) that would allow Macintosh computers to communicate through TCP/IP. One such gateway was initially developed at Stanford University. The client software encapsulated IP datagrams in AppleTalk DDP packets, which were then sent to a gateway. The gateway decapsulated the packets and forwarded them to the TCP/IP internet. The Stanford work also specified the inverse: encapsulation of AppleTalk DDP data in IP (UDP) packets. A networking company called Kinetics used the Stanford technology in a gateway product. Kinetics called the gateway a *Kinetics IP,* or KIP, [4] gateway. Today, KIP continues to be one of the primary implementations for providing a gateway between DDP and IP. The *Columbia AppleTalk Package* (CAP) [4] is another primary implementation for encapsulating DDP in UDP. The two types of encapsulation, IP in DDP (referred to as MacIP or DDP/IP) and DDP in UDP (referred to as IPTalk), provide distinct ways of integrating AppleTalk nodes in an IP environment.

MacIP allows AppleTalk nodes without direct IP connectivity to have access to a TCP/IP internet nevertheless. The MacIP gateway, both an AppleTalk node and an IP host, has a configured-client range of IP addresses and can assign one of its IP addresses dynamically to an AppleTalk node. The gateway maintains a mapping between DDP node addresses and their assigned IP addresses. IP hosts on the TCP/IP internet perceive the AppleTalk nodes to be ordinary IP hosts. The MacIP gateway creates the perception by acting as a proxy for the AppleTalk nodes. When the MacIP gateway receives an *Address Resolution Protocol* (ARP) [5] packet for an IP address assigned to a MacIP client, it answers the request with its own physical address. This action is called *proxy ARPing*. Packets destined for the MacIP client will thus be physically addressed to the MacIP gateway, which forwards the packets to the correct AppleTalk node.

ARP functionality is performed on the AppleTalk internet through NBP. Each MacIP client registers with an NBP name containing its own IP address and responds to NBP Lookups for its IP address. When a MacIP client wishes to send a packet to another IP host, it tries to locate the IP host by issuing an NBP Lookup for the IP address. If the Lookup is for another MacIP client managed by the MacIP gateway, the specified client responds directly to the Lookup. Otherwise, the MacIP gateway acts as a proxy for the other IP hosts. When the gateway receives NBP Lookups for any IP addresses that are outside its client range, it responds to the Lookup with its AppleTalk address.

When the requesting MacIP client receives the Lookup response, it learns the destination AppleTalk address for the IP packet (encapsulated in DDP) that it is sending. The gateway's NBP proxying ensures that the MacIP clients send the gateway any IP packets destined for the TCP/IP internet.

To communicate with services on a TCP/IP internet, an AppleTalk node sends IP data encapsulated in DDP to a MacIP gateway. The gateway strips off the DDP headers and then sends the packet through the TCP/IP internet. For packets going from TCP/IP to AppleTalk, the gateway encapsulates the IP data in DDP, addresses the packet to the AppleTalk node corresponding to the IP destination address, and sends the packet out. By handling dynamic IP address assignment, address resolution, and packet encapsulation, MacIP enables AppleTalk nodes to fully participate on a TCP/IP internet.

IPTalk allows AppleTalk nodes to communicate by way of an existing TCP/IP internet. Instead of tunneling through the IP internet with point-to-point connections as in AURP, IPTalk treats the IP internet as a virtual AppleTalk network. The IP network on which an IPTalk host resides has an assigned AppleTalk network number. The IPTalk host is a node on this AppleTalk network. Less dynamic than AppleTalk, IPTalk relies upon static, central administration of routes within the TCP/IP internet. IPTalk employs its own protocol to distribute routing information. Rather than using update-based propagation of routing information as in AURP, IPTalk relies initially on configured routing information and thereafter on periodic (once-a-minute) updates. IPTalk hosts rely on the IPTalk version of ZIP query and reply packets to obtain zone information.

An IPTalk gateway maintains a mapping between the IPTalk hosts' AppleTalk addresses and their corresponding IP addresses. The gateway also maintains IP routes (routes to access other IPTalk hosts or gateways) in addition to any local AppleTalk routes. It establishes IP routes both statically (at configuration time) and dynamically (from routing information learned from other gateways). To communicate to a node on the other side of a TCP/IP internet, an end node on an AppleTalk network sends an AppleTalk packet to an IPTalk gateway. The gateway then encapsulates the packet in UDP and sends the packet to the IPTalk host or gateway corresponding to the destination DDP network. The destination gateway, in turn, decapsulates the packet and sends it on to the destination DDP network. By using the TCP/IP internet as a virtual AppleTalk network, IPTalk thus enables AppleTalk nodes to send and receive packets across a TCP/IP internet.

Apple Remote Access

Apple Remote Access (ARA) [3] is a product developed by Apple Computer that allows remote users to dial into an AppleTalk internet (Figure 4). Through protocols specially designed to handle remote access via dial-up lines and routing techniques such as remapping and network hiding, ARA enables users to access remote AppleTalk network resources with virtually the same ease as accessing local resources.

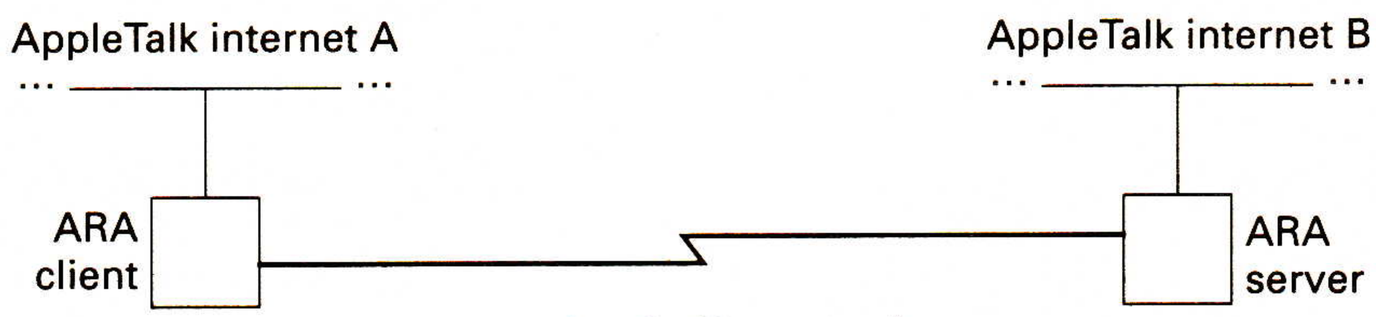

Figure 4: Apple Remote Access

continued on next page

AppleTalk Routing *(continued)*

When an ARA client connects to an ARA server, the client is able to participate on the server's network in addition to its own local network. ARA does not provide true routing or bridging between the two internets. Instead, the client maintains a unique vantage point. The other nodes on its local internet cannot see the nodes on the remote internet, and similarly the remote nodes cannot see the local nodes. In contrast, the client is able to see both internets and all the nodes on both networks can see the client. The client becomes a virtual node on the remote network in addition to being a node on the local network.

In the initial exchange of information, the server sends the client the list of zones accessible to the client in the remote internet (possibly filtering certain zones based on the client's authorization rights). The client's own zone name depends on the situation. If the client was already in a zone on its local internet, it maintains that zone name. Otherwise, it joins the zone of the remote network. In either case, remote nodes perceive the client to be in the server's zone, given that it is a virtual node on the server's network.

In the initial exchange the server also sends the network number and node ID that the client should use when sending AppleTalk packets. ARA resolves any network-number conflicts between the client and the server's network by remapping the two network numbers. ARA provides only remapping between the client and server networks. ARA eliminates other conflicts by hiding from the client any other remote-network numbers that conflict with those in the client's local internet. As a result, the services on the conflicting remote networks are not available to the client. (The client must disconnect from its local network to access those services.) ARA's method of resolving network conflicts places priority on the consistency of the client's internet.

Because the client clearly must send all packets destined for the remote internet to the server, which in turn forwards the packets, the client has no need for any RTMP routing information. As a result, the ARA server filters RTMP packets and does not forward them to the client. In addition, ARA compensates for the relatively slow link speed of dial-up lines by requiring a reliable connection and by using a technique called *smart buffering*. Smart buffering is a method for reducing line traffic by using tokens. Because retransmission of data can unnecessarily consume bandwidth over a reliable link, ARA does not send duplicate packets or even duplicate parts of packets. Instead, an ARA client sends tokens to represent the retransmitted, duplicate information, substantially reducing the amount of data sent.

ARA handles NBP similarly to routers. When the client needs to look up a network service, it sends an NBP *Broadcast-Request* (BrRq) packet, regardless of whether a router exists. When the ARA server receives the NBP BrRq packet, it treats it much as a router would. It determines whether to forward the lookup to a router or whether to send it out on its local network. Thus, through a combination of routing and forwarding techniques, an ARA server creates a virtual node on its network for every connected ARA client. Without providing full routing or bridging, the ARA server provides connectivity between remote nodes and the server's AppleTalk internet. The remote node, the ARA client, thus gains full access to the server's internet while maintaining access to its own local internet.

Summary

The AppleTalk suite of protocols supports routing in large data networks and has been designed for "plug-and-play" operation. AppleTalk not only provides dynamic routing for data messages but also dynamic name and address assignment for end nodes. With AppleTalk, network connection of an end node such as a computer or peripheral device requires almost no preconfiguration. Moreover, communication with that end node may begin almost immediately after it is connected to the AppleTalk network. AppleTalk includes protocols for binding end-node names and addresses, distributing network reachability and naming information among routers, and forwarding data messages between end nodes. It also includes protocols to support communication in multiprotocol internetworks. Specifically, AppleTalk employs tunneling for connecting noncontiguous AppleTalk networks, gateways for communicating between AppleTalk and TCP/IP networks, and remote access facilities for connecting users to distant AppleTalk networks over point-to-point links. Its ease of use combined with its ability to function in multiprotocol environments has made AppleTalk one of the most popular routing systems for corporate and academic internetworks.

References

[1] Andrews, Richard F., Oppenheimer, Alan B., Sidhu, Gursharan S., *Inside AppleTalk,* 2nd ed., Addison-Wesley, 1990.

[2] Apple Computer, Inc., "AppleTalk Update-Based Routing Protocol: Enhanced AppleTalk Routing," Available through APDA. Also available as RFC 1504, August 1993.

[3] Apple Computer, Inc., "AppleTalk Remote Access Protocol Version 1.0," Available in the *AppleTalk Remote Access Developer's Toolkit v. 1.0,* through APDA.

[4] P. Budne, "KIP AppleTalk/IP Gateway Functionality," Internet Draft.

[5] D. Comer, *Internetworking with TCP/IP,* 2nd ed., Vol. I, Prentice Hall, 1991.

[6] A. Tanenbaum, *Computer Networks,* Prentice Hall, 1981.

[7] Tittel, E., "Back to Basics: AppleTalk," *ConneXions,* Volume 9, No. 7, July 1995.

ALAN B. OPPENHEIMER graduated from M.I.T. in 1983 and immediately went to work for Apple Computer as a member of the team working on AppleTalk, the network for the soon-to-be-shipped Macintosh computer. Alan was responsible for the design of many of the pieces of the AppleTalk network system, and is a co-author of the book *Inside AppleTalk.* Alan led the effort to create AppleTalk Phase 2, which was introduced in June of 1989, and designed AURP, a protocol which is used to connect AppleTalk networks over the Internet. Alan also led the team at Apple responsible for Apple Remote Access. Wanting to bring AppleTalk's ease-of-use to the Internet and World-Wide Web, Alan left Apple to found Open Door Networks, Inc. in January of 1995. Most recently at Open Door, Alan created the WebDoor automatic Web publishing system. E-mail: `alan@opendoor.com`

FIDELIA KUANG holds a B.S. and M.S. in electrical engineering from Stanford University. She has worked for Apple Computer, Inc. since 1987 on a variety of networking products, including the Apple Internet Router, Apple IP Gateway, and Apple Remote Access. While working on the Apple Internet Router, she helped implement AURP, a protocol that enables tunneling of AppleTalk through TCP/IP. She is currently an engineering manager in Apple's Communications Technology group, which is responsible for network infrastructure products and Apple's Open Transport networking and communications architecture. Her Internet e-mail address is: `kuang@apple.com`

[This article is adapted from *Routing in Communication Networks* by Martha Steenstrup (editor), ISBN 0-13-010752-2, Prentice-Hall, 1995. Used with permission. —Ed.]

Street sweeping the Information Super Highway

by "Michael Underwood"

Despite being in pre-retirement these days, I do still do a bit of traveling along the perhaps over-touted *Information Super Highway* (ISH). And despite the inaptness of the ISH metaphor, I do notice the need for a metaphorical street sweeper of/on it:

Spamming

One of the few mailing lists/newsgroups I still belong to recently was the victim of a singularly inappropriate mass-mailing of the sort that's apparently called a "spamming" in the neo-vernacular, in (dis)-honor of some dolt(s) who occasionally mass mail messages expressing distaste for a particular meat product that's actually remembered with some fondness by those who lived through food rationing during World War II, because it was one of the few meats available then. Annoying enough, but, sadly, not uncommon. What was especially annoying in this case, though, is that half a dozen or more other dolts started discussing the offending message in all apparent seriousness "on" all the list/groups it shouldn't have been sent to in the first place. Indeed, as I write this, messages are still being sent on the (trust me, muddled and irrelevant) topic/thread.

Response

So I sent a message "off-list" to an old friend I knew to have kept more current than I, asking if somebody had taken the trouble to write a program that disguised the true sender's identity (the Protocol Police comes to mind for a pseudonym) and re-sent the offending message back to its sender ten-fold or a hundred-fold, after a line that said something like "If you insist on wasting our time and money, we'll return the disfavor." He didn't know of one, and, being more "philosophical" than I as well as more current, opined that this sort of thing happened and was best ignored. (He also emphasized that such a program would possibly lead to retaliatory "mail-bombs," which *is* why I'd assumed a false ID would be used—and is why I'm using a pseudonym for this piece.) While doubtless commendable, I found this laissez faire approach unsatisfying, as well as unsatisfactory, however.

A possible solution?

In thinking things over, I realized that it is remotely possible that however satisfying it might be, the "eye for an eye" approach I'd initially conceived of might, after all, be somewhat beneath one's dignity. Not that I'd fault anyone who chose to employ it, of course, but a somewhat more measured solution did occur to me fairly rapidly, and I thought I'd pass it along for what it's worth. (Clearly, it's not the only possible approach, and it may not even be a terribly good one, but at least it might get people stirring their stumps on other, better approaches.)

There does exist "list manager" software of some sort, I'm fairly confident, even if I don't know the details. Why not add a feature to it that would check incoming messages' senders (and IP addresses, just to make it somewhat harder to spoof) against the registered subscribers, and return a canned "Messages to this List/Group may be sent by registered subscribers only. To register for this List/Group, [do whatever]" instead of sending the unsolicited message out to everybody?

Well, I can think of a couple of objections all by myself, come to think of it: It *would* cost some microseconds, once implemented. Big deal. But it would also preclude legitimate contributions by "indirect subscribers," who see the List/Group through one or another concentrator/redistributor mechanism.

Worth the effort?

However tempting it is to be flip and wonder whether that isn't a smaller deal than the per-message microseconds, I'll content myself with observing that it shouldn't be hard to add an indirect subscriber registration process to the mix, for those List/Groups the owners of which choose to cater to submissions from indirect subscribers. (If, by the way, the insurmountable problem turns to be that the operating system of choice for list manager software doesn't make the IP addresses available to user-level programs, let's just observe, moderately, that we already knew it was in need of fixing anyway, so why not get on with it, finally?)

Now, the question remains whether it's worth the effort. Granted, we're not talking "hazmat" here. It isn't radioactive wastes or Agent Orange spills on the ISH, after all. Still, it does seem worth *some* trouble to sweep the fertilizer away, doesn't it?...

"**MICHAEL UNDERWOOD**" assumes that any of the Old Network Boys who want to discuss this matter with him "off-list" already know how to get in touch with him, or have the sense to ask Ole to forward their messages, and that nobody who's read this would even think of trying to tell him that "self-expression is important to the young." He adds that orange marmalade, cloves, and cinnamon baked with that product which was one of the only meats one could get made it rather palatable.

Write to *ConneXions*!

We'd love to hear your comments, suggestions and questions about anything you read in *ConneXions*. Our editorial address is given below. Use it for letters to the Editor, requests for the index of back issues, questions about particular articles etc.:

ConneXions—The Interoperability Report
303 Vintage Park Drive
Suite 201
Foster City
California 94404–1138
USA
Phone: +1 415-578-6900 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-525-0194
E-mail: `connexions@interop.com`
URL: `http://www.interop.com`

Subscription information

For questions about your subscription please call our customer service hotline: 1-800-575-5717 or +1 610-892-1959 outside the USA. This is the number for our new subscription agency, Seybold Publications. Their fax number is +1 610-565-1858. The mailing address for subscription payments is: P.O. Box 976, Media, PA 19063–0976.

Book Review

Networks by Timothy Ramteke, ISBN 0-13-958059-X, Prentice-Hall, 1994.

Introduction

As the field of networking has grown, rapidly, broadly and deeply, survey books have become increasingly difficult to write and essential to read. More newcomers need the introduction and more old-timers need to have at least passing familiarity with concepts and detail peripherally related to their daily work. So I approached *Networks* with considerable enthusiasm. A quick survey of the content I could evaluate easily left me confused. Eventually, I *did* note that the title was "Networks" and not "Networking." This is a book primarily about telecommunications technology and not data networking. The author thinks otherwise and even suggests ways to use the book for WAN or LAN courses.

Bad news

So, here's the bad news and then I'll suggest why I'm glad to have the book around: *Networks* simply does not cover datagram-based networking very well. For a book published in 1994, it's odd that the author was unaware that DARPA hasn't been a major funder of the Internet for many years, makes no mention of NSFNET, and doesn't know that the ARPANET ceased to exist in 1990. The discussion on TCP acknowledgment and windowing is a good effort, but doesn't quite work. For a book introducing technical concepts, it's also odd that there are quite a few pages discussing UNIX commands for Internet activity. The chapter on NetWare also shows user interactions. The chapter called "Interconnecting LANs" combines bridging and routing with discussion of the newer, high-speed transmission technologies (e.g., Frame Relay and ATM.) The author cites assistance he received from the eminent Radia Perlman and the discussion on bridging and routing is, in fact, OK. Just OK. Don't design any backbones with this as your only reference material.

Good news

Now the good news. The book starts with a section on history, dating back to 1831. The chapter on "Analog and Digital Signals" opens with a description of Ohm's Law! Various telephone transmission and switching technologies, e.g., Centrex and Signalling System 7, are covered as are several national telephone service architectures. This is fun stuff and quite eclectic. It tends to be primarily expository description of functional characteristics. Underlying detail and theory is often lacking, as is critical analysis. But the compendium of detail was easy to read and satisfied some long-standing curiosity I've had about the part of the world that is activated when I pick up a handset.

Party line

Part I, Fundamentals, is 50 pages covering history, abstract architectures, basic signals and transmission technologies for the signals to travel over. In case you don't have an OSI 7-layered picture around, the one in *Networks* shows X.25 at the network layer, with no discussion of the layer's subdivision into components for internetworking. The application layer exemplars are, of course, X.400 and X.500. In other words, the author attempts to carefully tow the party line in the book's chapters.

If you require clever personal asides ranging from soap boxes pronouncements to flaming criticism, this book will leave you unfulfilled. If you want simple and clear exposition, this book is quite comfortable. Part II, Voice Networking, is 155 pages, with chapters covering signalling and switching, public and private networks, call processing mechanisms, T1-speed systems and virtual networking.

The 110 pages of Part III, Wide Area Networks, covers SNA, X.25, SS7, ISDN and SONET. So even in this section the telephone company focus persists. The final part is on LANs and Internetworking. Its 140 pages span LANS—primarily access methods—NetWare, interconnection, and TCP/IP, as discussed above.

Recommended

So who *should* read this book? I'd recommend it for a datacom person with no background on the phone technology. Same for programmers light on the underpinnings for communications infrastructure.

—D. Crocker, Brandenburg Consulting
`dcrocker@brandenburg.com`

Future NetWorld+Interop Dates and Locations

The topics covered in *ConneXions—The Interoperability Report* are also discussed at the NetWorld+Interop conference and exhibition. In 1996 you will have no less than seven opportunities to attend the show. Our calendar of events is given below.

NetWorld+Interop 95	Paris, France	September 11–15, 1995
NetWorld+Interop 95	Atlanta, GA	September 25–29, 1995
NetWorld+Interop 96	Las Vegas, NV	April 1–5, 1996
NetWorld+Interop 96	Frankfurt, Germany	June 10–14, 1996
NetWorld+Interop 96	Tokyo, Japan	July 15–19, 1996
NetWorld+Interop 96	Atlanta, GA	September 16–20, 1996
NetWorld+Interop 96	Paris, France	October 7–11, 1996
NetWorld+Interop 96	London, England	Oct 28–Nov 1, 1996
NetWorld+Interop 96	Sydney, Australia	November 25–29, 1996

All dates are subject to change.

Call 1-800-INTEROP or 1-415-578-6900 for more information. Or send e-mail to `info@interop.com` or fax to 1-415-525-0194. For the latest information about NetWorld+Interop including *N+I Online!* as well as other SOFTBANK produced events, check our home page at `http://www.interop.com`

NetWorld+Interop is produced by SOFTBANK Exposition and Conference Company, 303 Vintage Park Drive, Foster City, California 94404–1138, USA.

NETWORLD+INTEROP

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

EDITOR and PUBLISHER Ole J. Jacobsen